Datasets

A dataset combines the vector (feature) representation of objects with labels (class names), prior probabilities and possible other data annotation. a, b and d are datasets. Almost all PRTools commands expect data to be supplied as a PRTools dataset.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

b = a*(w*) map b by w
w = [w1 w2 ...]*w1 w2 ... stacking of modules
w = [w1;w2;...]*w1 w2 ... parallel combination
w = [w1 w2 ...]*maxc w1 w2 ... maxc, such that maxc is the output of the individual classifiers are combined by the maxc rule. More many exist.

Datasets

A datafile is a pre-stage of a dataset. It refers to objects organized as files (e.g. images) in directories and stores all pre-processing and feature definition needed to convert it to a dataset. Datasets may bring large amounts of raw data (not yet normalized, varying sizes, no features extracted) within the domain of PRTools. Many commands defined for datasets apply to datafiles as well.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults

Datafiles

A datafile is a pre-stage of a dataset. It refers to objects organized as files (e.g. images) in directories and stores all pre-processing and feature definition needed to convert it to a dataset. Datasets may bring large amounts of raw data (not yet normalized, varying sizes, no features extracted) within the domain of PRTools. Many commands defined for datasets apply to datafiles as well.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults

Datafiles

A datafile is a pre-stage of a dataset. It refers to objects organized as files (e.g. images) in directories and stores all pre-processing and feature definition needed to convert it to a dataset. Datasets may bring large amounts of raw data (not yet normalized, varying sizes, no features extracted) within the domain of PRTools. Many commands defined for datasets apply to datafiles as well.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults

Datafiles

A datafile is a pre-stage of a dataset. It refers to objects organized as files (e.g. images) in directories and stores all pre-processing and feature definition needed to convert it to a dataset. Datasets may bring large amounts of raw data (not yet normalized, varying sizes, no features extracted) within the domain of PRTools. Many commands defined for datasets apply to datafiles as well.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults

Datafiles

A datafile is a pre-stage of a dataset. It refers to objects organized as files (e.g. images) in directories and stores all pre-processing and feature definition needed to convert it to a dataset. Datasets may bring large amounts of raw data (not yet normalized, varying sizes, no features extracted) within the domain of PRTools. Many commands defined for datasets apply to datafiles as well.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults

Datafiles

A datafile is a pre-stage of a dataset. It refers to objects organized as files (e.g. images) in directories and stores all pre-processing and feature definition needed to convert it to a dataset. Datasets may bring large amounts of raw data (not yet normalized, varying sizes, no features extracted) within the domain of PRTools. Many commands defined for datasets apply to datafiles as well.

Sampling datasets or datafiles

b = a(objects,:) get subset of objects
b = a(:,features) get subset of features (no datafiles)

Mapping definition (in scripts and functions)

A mapping stores the definition of a mapping of one object representation (e.g. a vector space) into another. Some mappings may be trainable: they can be optimized for a given dataset.

w = mapping(file,type,data, ...) low level routine to define mapping
args = setdefaults(argin,def1,...) set defaults in mapping routine
w = define_mappings(args,type) high level routine to define mapping in combination with setdefaults
## PRTools Procedures

### Data generation
- `circles3d, lines5d` - 3D circles and lines
- `gendatb, gendatc, gendatd, gendatf, gendatg` - generation of subsets
- `gendatm, gendatspirals` - 2D problems
- `gendet, gendetw, gen subsets` - data generation
- `gendatk, gendatp` - interpolation

### Data import
- `prdata` - load raw data, convert to dataset
- `prdataset` - load dataset from matfile
- `prdatasets` - load public domain data
- `prdatafiles` - load public domain data as dataset

### Handling images
- `data2im` - convert dataset to image
- `obj2feat, feat2obj` - object images <--> feature images
- `im2feat, im2obj` - image to feature or pixel classification
- `imsave` - retrieve size of specific image in dataset
- `im_patch` - find / generate patches in object images
- `bands2obj` - convert image bands to objects in dataset
- `bandsel` - select image bands in dataset or datafile
- `selectim` - select image in multi-band object image

### Image operations
- `classim` - classify image using a given classifier
- `doublem` - convert datafile images into double
- `filtim` - filter images for datafiles and datasets
- `spatm` - spatial smoothing of pixel classification
- `datunif, datgauss, im_box, im_fft, im_gray, im_label, im_maxf, im_minf, im_norm, im_resize, im_rotate, im_scale, im_select_blob, im_threshold`
- `histm, im_harris, im_moments, im_mean, im_measure, im_profile, im_stat, im_skel_meas`

### Features from images
- `featval` - evaluation of a feature set
- `featran` - ranking of individual feature performances
- `featsel` - user supplied feature selection

### Fixed mappings
- `cmapm` - some special maps
- `sigm, invsigm` - (inverse) sigmoid map
- `filtm` - arbitrary operation on datafiles/datasets
- `normm` - object normalization
- `remout` - remove outliers

### Trainable mappings
- `scalem` - find appropriate scaling
- `bhtm, fisherm, chernoffm, nlfisherm` - linear supervised mappings
- `klm, kls` - decorrelation and Karhunen Loève mapping
- `pca` - principal component analysis
- `proxm` - proximity mapping and kernel construction
- `reducem` - reduce to minimal space mapping
- `kernelm` - kernel mapping
- `userkernel` - user supplied kernel definition
- `gtd, som` - special mappings

### Density estimation
- `gaussm` - mixture of Gaussians
- `knnm` - k-Nearest neighbor density
- `parzenm` - Parzen density
- `parzenml` - ml estimation of smoothing for Parzen

### Clustering and distances
- `dism` - distance matrix between two datasets.
- `emclust` - expectation - maximization clustering
- `proxm` - proximity mapping and kernel construction
- `hclust` - hierarchical clustering
- `kcentres` - k-centres clustering
- `kmeans` - k-means clustering
- `mode seek` - clustering by mode seeking
- `mds, mds_cs` - multi-dimensional scaling

### Regression
- `linearr, ridger, lassos, svmr, ksmoothr, knnr, pinvr, pls, plsgrp, gpr, trs, rsquard, gendatf`

### Classifiers, linear and quadratic
- `fisherc, ldc, logic, mnc, mncs, qdc, udc`
- ` diplom, nullibsvm, rlibsvm, pllibsvm` - based on the LIBSVM package
- `svc, nusvc, rbsvc` - PRTTools based SVM

### Classifiers, support vector machine (svm)
- `bpxnc, lmnc, perlc, rbnc, rnc, vpc, drbmc`
- `mogc, parzenc, parzencd, mnc, ldc, udc, qdc, naivebmc, density based classifiers`
- `treec, dtc, randomforest, stumpc` - decision trees
- `weakc, knnc, baggingc, ada boostc, fdc` - other classifiers

### Classifiers, various
- `avggc, dgc, modselc, rssoc, votec, wvotec, maxc, mnc, mncs, medianc, mllc, naivebmc, perc, prodc, traincc`

### Combining classifiers
- `avggc, dgc, modselc, rssoc, votec, wvotec, maxc, mnc, mncs, medianc, mllc, naivebmc, perc, prodc, traincc`
- `distmah` - Mahalanobis distance more routines
- `mean cov` - Estimation of means and covariances
- `edicon` - Edit and condense training sets
- `testk` - Error estimation for k-nearest neighbour rule
- `testp` - Error estimation for Parzen classifier
- `testn` - Error estimate for normal distributions
- `testc` - General error estimation routine
- `classc` - Converts a mapping into a classifier
- `labeld` - Find labels of objects by classification
- `rejectc` - Creates reject version of existing classifier

### Evaluation
- `classim` - classify image using a given classifier
- `clevale` - classifier evaluation (feature size curve)
- `confmat` - computation of confusion matrix
- `costm` - cost mapping, classification using costs
- `crossval` - crossvalidation
- `disperror` - display annotated error matrix
- `labelim` - construct image of labeled pixels
- `lodo` - leave_one_set_out crossvalidation
- `reject` - compute error-reject trade-off curve
- `roc` - receiver-operator curve (ROC)
- `shiftop` - shift operating point of classifier
- `testc` - general classifier error estimation routine
- `testd` - error of dataset applied to given classifier
- `testauc` - estimate error as area under the ROC

### Plot routines
- `plotc, plotm` - plot classifier, mapping in scatterplot
- `plote` - plot error curves
- `plotf` - plot feature distribution
- `ploto` - plot object functions
- `plotdg` - plot dendrogram (see hclust)

### Examples
- `prex_clevale, prex_combining, prex_confmat, prex_datafile, prex_datasets, prex_density, prex_eigenfaces, prex_matchlab, prex_mctxt, prex_plotc, prex_som, prex_spatm, prex_cost, prex_logdens, prex_soft, prex_regr`

---